Fair versus Unrestricted Bin Packing *

Yossi Azar † Joan Boyar ‡ Leah Epstein §

Lene M. Favrholdt ‡ Kim S. Larsen ‡ Morten N. Nielsen ‡

Corresponding author:

Joan Boyar, Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark, joan@imada.sdu.dk.

Phone : +45 6550 2338. Fax: +45 6593 2691.

Keywords: On-line algorithms, Competitive analysis, Bin packing, Dual bin packing, Restricted adversaries, Randomization, Admission control.

---

*A preliminary version of this paper appeared as: Yossi Azar, Joan Boyar, Lene M. Favrholdt, Kim S. Larsen, Morten N. Nielsen. "Fair versus Unrestricted Bin Packing". *Proceedings of the Seventh Scandinavian Workshop on Algorithm Theory*, Lecture Notes in Computer Science, vol. 1851, pages 200–213, Springer-Verlag, 2000.

† Department of Computer Science, Tel-Aviv University, Israel, azar@tau.ac.il. Supported in part by the Israel Science Foundation, and by a USA-Israel BSF grant.

‡ Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark, {joan,lenem,kslarsen,nyhave}@imada.sdu.dk. Supported in part by the Danish Natural Science Research Council (SNF) and in part by the IST Programme of the EU under contract number IST-1999-14186 (ALCOM-FT).

§ School of Computer and Media Sciences, The Interdisciplinary Center, Herzliya, Israel, lea@idc.ac.il.

Abstract

We consider the on-line Dual Bin Packing problem where we have n unit size bins and a sequence of items. The goal is to maximize the number of items that are packed in the bins by an on-line algorithm. We investigate *unrestricted* algorithms that have the power of performing admission control on the items, i.e., rejecting items while there is enough space to pack them, versus *fair* algorithms that reject an item only when there is not enough space to pack it. We show that by performing admission control on the items, we get better performance compared with the performance achieved on the fair version of the problem. Our main result shows that with an unfair variant of First-Fit, we can pack approximately 2/3 of the items for sequences for which an optimal off-line algorithm can pack all the items. This is in contrast to standard First-Fit where we show an asymptotically tight hardness result: if the number of bins can be chosen arbitrarily large, the fraction of the items packed by First-Fit comes arbitrarily close to 5/8.

1 Introduction

The Problem Bin Packing is one of the most classical problems in combinatorial optimization and in theoretical computer science. In the *Classical Bin Packing problem* we are given an unlimited number of unit bins and a set of items each with a non-negative size where the goal is to minimize the number of bins used to pack all the items. In the *Dual Bin Packing problem*, we are given a fixed number n of unit size bins and a set of items each with a non-negative size where the goal is to maximize the number of items packed. The Dual Bin Packing problem has been studied in the off-line set-ting, starting in [8], and its applicability to processor and storage allocation is discussed in [9]. (For surveys on Classical Bin Packing, see [7, 10].) In the on-line version of the problem, the items arrive in some sequence and the assignment of an item should be done before the next item arrives.

In this paper, we consider the on-line *Dual Bin Packing problem*. In this problem, the algorithm may not be able to pack all items and the question is whether the algorithm is allowed or not allowed to perform admission control. In [6], the *Fair Bin Packing* problem was investigated. In Fair Bin Packing, an algorithm is only allowed to reject an item if it cannot fit in any bin at the time when it is given. Note that, for this version of the problem, the off-line algorithm is also required to be fair. In this paper, we also consider what happens when the fairness restriction is removed and call the problem *Unrestricted Bin Packing*.

The Performance Measures The standard measure for the quality of on-line algorithms is the *competitive ratio*. For the Bin Packing problem, the competitive ratio of an algorithm $\mathbb{A}$ is the worst case ratio, over all possible input sequences, of the number of items packed by $\mathbb{A}$ to the number of items packed by an optimal off-line algorithm.

For the Bin Packing problem, as well as for many other on-line problems, the competitive ratio yields very pessimistic results. In particular, for the maximization problems no algorithm can pack a constant fraction of the number of items packed by the optimal algorithm and the competitive ratio must depend on the size of the smallest item (see [1, 2, 6], for example). Since we are interested in results that hold for arbitrary size items (and get constant competitive ratios similar to most of the results in the Classical Bin Packing problem), we need to restrict the input sequences. Having in mind the Classical Bin Packing problem where all items are required to be packed, a natural assumption is to restrict the input sequences to those which can be completely packed by an optimal off-line algorithm. This enables us to obtain significantly better results. Such sequences are called *accommodating sequences*, since the off-line algorithm can accommodate the whole sequence.

Note that on accommodating sequences, the competitive ratio of Unrestricted Bin Packing is no worse than the competitive ratio of the fair problem, since the off-line algorithm packs all items and hence is fair. In general, however, the competitive ratio of Unrestricted Bin Packing is not necessarily better than the competitive ratio of the fair problem since also the off-line algorithm can benefit from not being fair. In fact, in many cases, considering unfair algorithms, i.e., performing admission control on the items, is the more challenging problem; see for example the results for throughput routing in [1, 2, 3]. In particular, with the Unrestricted Bin Packing problem, the competitive ratio of different algorithms can vary over a large range. This is in contrast to on-line algorithms for Fair Bin Packing where all competitive ratios for deterministic algorithms are within a constant factor of each other, both for arbitrary sequences and for accommodating sequences (see [6]).

The competitive ratio and accommodating sequences are defined formally in Section 2.

The Results The results in this paper are for accommodating sequences for the on-line Dual Bin Packing problem. In [6], it is shown that, for the Fair Bin Packing problem, First-Fit has a competitive ratio of at least $\frac{5}{8}$ on accommodating sequences, i.e., it packs at least $\frac{5}{8}$ of the items. In this paper, we show that the bound is asymptotically tight, i.e., the competitive ratio on accommodating sequences comes arbitrarily close to $\frac{5}{8}$ for large enough n.

More specifically for any n, the competitive ratio is bounded by $\frac{5}{8} + O(\frac{1}{\sqrt{n}})$.

Since the competitive ratio on accommodating sequences is no worse for the unfair version of the Bin Packing problem than for the fair version, First-Fit has a competitive ratio of $\frac{5}{8}$ also for Unrestricted Bin Packing. The main result in this paper is that we can do better. We design an algorithm called Unfair-First-Fit whose competitive ratio on accommodating sequences is about $\frac{2}{3}$. More precisely, it is $\frac{2}{3} \pm \Theta(\frac{1}{n})$. Hence, starting from some value of n, the competitive ratio of Unfair-First-Fit on accommodating sequences is strictly better than that of First-Fit.

We show that the competitive ratio of any on-line algorithm on accommodating sequences is no better than $0.857 + O(\frac{1}{n})$, even when considering randomized algorithms. For deterministic fair algorithms, we prove a slightly better hardness result of $0.809 + O(\frac{1}{n})$.

2 The Performance Measures

For completeness, we define the competitive ratio [14, 12] and accommodating sequences [5]. Note that Dual Bin Packing is a maximization problem, and all ratios are less than or equal to one.

Let $\mathbb{A}(I)$ denote the number of items algorithm $\mathbb{A}$ accepts when given the sequence I and let $\mathrm{OPT}(I)$ denote the number of items an optimal off-line algorithm, OPT, accepts.

Definition 2.1 An on-line algorithm, $\mathbb{A}$, is *c-competitive* if

$$\mathbb{A}(I) \geq c \cdot \mathrm{OPT}(I), \text{ for all input sequences } I.$$

The *competitive ratio* $CR = \sup\{c \mid \mathbb{A} \text{ is } c\text{-competitive}\}$. $\square$

Sometimes in the definition of the competitive ratio, an additive term is allowed, so the requirement is weakened to $\mathbb{A}(I) \geq c \cdot \mathrm{OPT}(I) - b$, where b is a fixed constant independent of I [4]. In that situation, our definition would then be referred to as the *strictly* competitive ratio. However, we will not need the additive term in this paper.

Furthermore, one could have chosen to focus on the inverse ratio to obtain numbers larger than one. However, we made our choice for consistency with similar decisions in the area of approximation algorithms where ratios for maximization problems are smaller than one and the inverse is referred to as the approximation factor [11].

Next, we define accommodating sequences.

Definition 2.2 A sequence of items is an *accommodating sequence* if an optimal off-line algorithm can pack the whole sequence using the n bins. □

Finally, we define the competitive ratio on accommodating sequences.

Definition 2.3 An on-line algorithm, $\mathbb{A}$, is *c-competitive on accommodating sequences* if

$$\mathbb{A}(I) \geq c \cdot \mathrm{OPT}(I), \text{ for all accommodating sequences } I.$$

The *competitive ratio on accommodating sequences* is

$$AR = \sup\{c \mid \mathbb{A} \text{ is } c\text{-competitive on accommodating sequences}\}.$$

□

3 Hardness Results

In this section, we show hardness results on the competitive ratio on accommodating sequences. Recall that when the sequences are all accommodating, an optimal off-line algorithm always behaves fairly. Hence, an algorithm for Fair Bin Packing is also an algorithm for Unrestricted Bin Packing with the same performance.

First, we show a negative result on the performance of First-Fit. This demonstrates the need for a better algorithm.

Theorem 3.1 For any n, First-Fit's competitive ratio on accommodating sequences is at most $\frac{5}{8} + O(\frac{1}{\sqrt{n}})$. If n is of the form $n = 9 \cdot 2^q - 5$, $q \in \mathbb{N}_0$, then First-Fit's competitive ratio on accommodating sequences is at most $\frac{5}{8} + \frac{1}{6n}$.

Proof We first show the bound for the special values of n. Then we extend it to all values of n. Let $\varepsilon > 0$ be small enough. An adversary can give the following sequence of items, divided into $q + 3$ phases:

Phase 1.	3 items of size $A = \frac{1}{3} - 2^{3q}\varepsilon$.
Phases $2 \ldots (q+1)$.	In phase $j+1$ $(1 \leq j \leq q)$, $3 \cdot 2^j$ pairs, each with one item of size $B_j = \frac{1}{3} + 2^{3q-3j+2}\varepsilon$ followed by an item of size $C_j = \frac{1}{3} - 2^{3q-3j}\varepsilon$.
Phase $q+2$.	$3 \cdot 2^q$ items of size $D = \frac{2}{3} + \varepsilon$.
Phase $q+3$.	$9 \cdot 2^q - 6$ items of size $E = \frac{1}{3}$.

First-Fit will pack the three items of the first phase in one bin, with three items. The assumption on ε assures that four items of size A cannot be packed together. For Phases $2, \ldots, q+1$, First-fit will pack one pair in each bin. For every Phase $j+1$, each packed bin will contain one item of size B_j and one item of size C_j using $3 \cdot 2^j$ bins. After such a pair is packed, all future items are too large to join a pair. The number of bins used in the first $q+1$ phases is $1 + \sum_{j=1}^{q} 3 \cdot 2^j = 6 \cdot 2^q - 5$. In the next phase, each item will be placed in its own bin, using the last $3 \cdot 2^q$ bins. There will be no space for items from the last phase.

OPT can pack each item from Phase 1 with two of the items of size B_1 from Phase 2, using a total of 3 bins for this. Then, it can combine two items of size B_{j+1} with one item of size C_j (for all $j \leq q-1$). This occupies $3 \cdot 2^j$ bins for all $1 \leq j \leq q-1$. Finally, it can pack one item of size C_q together with one item of size D using a total of $3 \cdot 2^q$ for this. The number of bins used is $3 + 3\sum_{j=1}^{q-1} 2^j + 3 \cdot 2^q = 6 \cdot 2^q - 3$. There are now $3 \cdot 2^q - 2$ empty bins which can each hold three items from the last phase. The ratio is thus $\frac{15 \cdot 2^q - 9}{24 \cdot 2^q - 15} = \frac{5}{8} + \frac{1}{8(8 \cdot 2^q - 5)} \leq \frac{5}{8} + \frac{1}{6 \cdot (9 \cdot 2^q - 5)} = \frac{5}{8} + \frac{1}{6n}$.

Now we extend the proof for arbitrary n. Let $n = n's + c$, where $n' = 9 \cdot 2^q - 5 = \Theta(\sqrt{n})$ and $q, s, c \in \mathbb{N}$ and $c < n'$. To get the hardness result against First-Fit, the adversary first gives c items of size 1 and then repeats s times each phase of the above sequence for n' bins. Clearly, First-Fit will accept $c + s(15 \cdot 2^q - 9)$ where OPT accepts $c + s(24 \cdot 2^q - 15)$, and thus the ratio is $\frac{5}{8} + O(\frac{1}{n'} + \frac{c}{n}) = \frac{5}{8} + O(\frac{1}{\sqrt{n}})$ □

In [6], it is shown that the competitive ratio of First-Fit on accommodating sequences is at least 5/8. Hence, 5/8 is an asymptotically tight bound on the competitive ratio of First-Fit on accommodating sequences.

In [6], it was shown that any deterministic Fair Bin Packing algorithm has a competitive ratio on accommodating sequences of at most $\frac{6}{7}$ for n even. The same result and essentially the same proof hold when the fairness restriction is removed, even for randomized algorithms.

Theorem 3.2 Any deterministic or randomized on-line algorithm for Unrestricted Bin Packing has a competitive ratio of at most $\frac{6}{7} + \frac{6}{21n-7}$, even on accommodating sequences.

Proof Let $\varepsilon > 0$ be a small enough constant. Consider an arbitrary on-line algorithm $\mathbb{A}$. An adversary can proceed as follows: Give n items of size $\frac{1}{2}-\varepsilon$, and let q denote the number of bins which contain two items after this. In the case where $E[q] < \frac{2n}{7}$, the adversary gives $\lfloor\frac{n}{2}\rfloor$ long items of size 1. The off-line algorithm can pack the first n items in the first $\lceil\frac{n}{2}\rceil$ bins and thus accept all items. On average, the on-line algorithm places two items in $E[q]$ bins and has at most one item in every other bin. Thus, by linearity of expectation, the performance ratio is at most $\frac{E[q]+n}{n+\lfloor\frac{n}{2}\rfloor} \leq \frac{E[q]+n}{n+\frac{n-1}{2}} < \frac{2n+4n/7}{3n-1} = \frac{18n}{21n-7} = \frac{6}{7} + \frac{6}{21n-7}$. In the case where $E[q] \geq \frac{2n}{7}$, the adversary gives n items of size $\frac{1}{2}+\varepsilon$. The off-line algorithm can pack the first n items one per bin and thus accept all $2n$ items. The on-line algorithm must reject at least $E[q]$ items on average. The performance ratio is at most $\frac{2n-E[q]}{2n} \leq \frac{6}{7}$. $\square$

However, for fair deterministic algorithms, we can slightly improve the hardness result of Theorem 3.2.

Theorem 3.3 The competitive ratio of any fair deterministic on-line algorithm is at most $\frac{(4\sqrt{3}-3)(n-1)}{(8\sqrt{3}-9)(n-1)-2} = \frac{23+4\sqrt{3}}{37} + O(\frac{1}{n}) < 0.809 + O(\frac{1}{n})$, even on accommodating sequences.

Proof Let $\varepsilon > 0$ be a small enough constant. Consider an arbitrary fair on-line algorithm $\mathbb{A}$. First, assume that n is even. An adversary can start the sequence by n items of size $1/2 - 2\varepsilon^3$. Let q denote the number of on-line bins containing two items. Since the algorithm is fair, all items are accepted. Hence, there are also q empty bins, and the remaining $n - 2q$ bins contain one item each. We continue by one of two different sequences, depending on the value of q. If $q \geq (2 - \sqrt{3})n$, we get the following sequence of items, containing five phases.

1. $n-q$ items of size $1/2+2\varepsilon^3$.
2. $q-1$ items of size $1/2-6\varepsilon^3$.
3. One item of size $1/2-8q\varepsilon^3+2\varepsilon^3$.
4. $q-1$ items of size $8\varepsilon^3$.
5. $2q-1$ items of size $4\varepsilon^3+\varepsilon^4$.

All items of Phase 1 are packed in an on-line bin which contains at most one item of size $\frac{1}{2}-2\varepsilon^2$. All items of Phases 2 and 3 are packed in a bin with one Phase 1 item. Denote the bin that got the item of Phase 3 by z. Note that since ε is small enough, the item of Phase 3 could only fit into a bin with one item. At this point, all on-line bins except z are filled to at least $1-4\varepsilon^3$. Hence, all items of Phase 4 fit into z, and do not fit into any other bin. Consequently, bin z is also occupied by $1-4\varepsilon^3$ after Phase 4. There is no room for any item of Phase 5.

OPT has $n-q$ bins with the pair $1/2-2\varepsilon^3$ and $1/2+2\varepsilon^3$, $q-1$ bins with the triplet $1/2-6\varepsilon^3$, $1/2-2\varepsilon^3$, $8\varepsilon^3$, and all other items in one bin. The total number of items is $2n+3q-2$ and the on-line algorithm accepts $2n+q-1$ of them. The ratio $\frac{2n+q-1}{2n+3q-2}$ grows as q decreases. Thus, the ratio is at most $\frac{2n+(2-\sqrt{3})n-1}{2n+3(2-\sqrt{3})n-2}=\frac{(4-\sqrt{3})n-1}{(8-3\sqrt{3})n-2}$.

If $q<(2-\sqrt{3})n$, we continue with the following five phases.

1. q items of size 1.
2. $n-2q-1$ items of size $1/2-4\varepsilon^2+2\varepsilon^3$.
3. One item of size $1/2+4\varepsilon^2-2\varepsilon^3-2\varepsilon^2(n-2q)(2-\varepsilon)$.
4. $n/2-q-1$ items of size $8\varepsilon^2-4\varepsilon^3$.

5. $n - 2q - 2$ items of size $4\varepsilon^2 + \varepsilon^4$.

After the first phase, the on-line algorithm has no empty bins. All items of Phase 2 are packed in bins with a single item of the initial sequence. Since ε is small enough, the item of Phase 3 is also packed in such a bin; denote this bin by w. All items in Phase 4 are also packed in bin w. There is again no room for items of Phase 5.

OPT has q bins with one item of size 1, $n/2$ bins with two items of the initial phase, $n/2-q-1$ bins with a pair of items from Phase 2 and one from Phase 4, and all other items in one bin. The total number of items is $3.5n - 4q - 3$, and the on-line algorithm accepts $2.5n - 2q - 1$. The ratio $\frac{2.5n-2q-1}{3.5n-4q-3}$ grows when q increases. Thus, the ratio is at most $\frac{2.5n-2(2-\sqrt{3})n-1}{3.5n-4(2-\sqrt{3})n-3} = \frac{(4\sqrt{3}-3)n-2}{(8\sqrt{3}-9)n-6}$, which is larger than the bound from case 1.

If n is odd, the adversary gives an item of size 1 just before Phase 1. This item will be accepted by both algorithms. Then the five phases described above are given with $\frac{n}{2}$ replaced by $\lfloor\frac{n}{2}\rfloor$. The performance ratio is at most $\frac{2.5(n-1)-2(2-\sqrt{3})(n-1)-1+1}{3.5(n-1)-4(2-\sqrt{3})(n-1)-3+1} = \frac{(4\sqrt{3}-3)(n-1)}{(8\sqrt{3}-9)(n-1)-2} = \frac{23+4\sqrt{3}}{37} + \frac{8\sqrt{3}-6}{(273-144\sqrt{3})n+128\sqrt{3}-255} < 0.809 + O(\frac{1}{n})$. This is the weakest of the bounds, and thus the result. $\square$

4 Unfair-First-Fit

4.1 The Algorithm

The algorithm Unfair-First-Fit (UFF), presented below, is shown to have a competitive ratio on accommodating sequences which approaches $\frac{2}{3}$ as n increases. Hence, above some fixed n, UFF is strictly better than First-Fit.

```
Input:   S = ⟨o_1, o_2, ... , o_n⟩
Output:  A, R, and a packing for those items in A
A := {o_1}; R := {}; S := tail(S)
while S ≠ ⟨⟩
    o := hd(S); S := tail(S)
    if size(o) > 1/2 and |A|/(|A|+|R|+1) ≥ 2/3
        R := R ∪ {o}
    else if there is space for o in some bin
            pack o according to the First-Fit rule
            A := A ∪ {o}
        else
            R := R ∪ {o}
```

Figure 1: The algorithm Unfair-First-Fit

In the description of Unfair-First-Fit (see Figure 4.1), A denotes the set of items accepted and R denotes the set of items rejected. Since every item is worth the same, it seems reasonable to reject large items. Therefore, for each item in the input sequence, Unfair-First-Fit examines if the item is larger than $\frac{1}{2}$ and if the performance ratio would still be at least $\frac{2}{3}$, even if the item were rejected. If both conditions are satisfied, the item is rejected (placed in the set R); otherwise the item is accepted (placed in the set A). All accepted items are packed according to the First-Fit packing rule.

4.2 Competitive Ratio on Accommodating Sequences

We introduce some notation: As stated above, A denotes the set of items accepted by Unfair-First-Fit, and R denotes the items rejected by Unfair-First-Fit. The term "large" is used for items strictly larger than $\frac{1}{2}$, since they are considered in a special way by the algorithm. Let L denote the set of large items that are alone in a bin in UFF's packing. Let s denote the size of the smallest item in R. We divide R into two disjoint sets, R_s containing small items and R_l containing large items. Let t denote the time just after the last large item was accepted by UFF and let A_t denote the set of items accepted at time t. We assume that the bins are numbered from 1 through n and ordered from left to right. When First-Fit is used to pack a sequence of items, it uses the bins in that order.

We now show that Unfair-First-Fit has a competitive ratio of approximately $\frac{2}{3}$. Since we consider accommodating sequences, all items will be packed by an optimal off-line algorithm. Therefore the sum of the sizes of items which are rejected is at most the sum of the empty space in all bins in Unfair-First-Fit's packing. We show later that at most n items can be rejected. Therefore a sequence showing that $\frac{2}{3}$ is impossible must be packed by Unfair-First-Fit with some items alone. We show that to have a ratio smaller than $\frac{2}{3}$, the average empty space in each bin is approximately $\frac{1}{3}$. With many large items rejected, the total number of rejected items is significantly smaller than n, which is the intuition behind the algorithm. We first relate the number of items packed alone to the number of (large) items rejected by admission control, and, finally, we relate this to the total number of items rejected.

Theorem 4.1 For $n \geq 9$, the competitive ratio of Unfair-First-Fit on accommodating sequences is more than $\dfrac{2}{3} - \dfrac{4}{6n+3}$.

Proof We divide the proof into two cases depending on the size of s. The first case is easy.

Case 1: $s > \frac{1}{2}$: Since the smallest item in R is larger than $\frac{1}{2}$, the items in $R \cup L$ are all larger than $\frac{1}{2}$. Thus, since all items can be packed in n bins, $|R| + |L| \leq n$, or $|R| \leq n - |L|$. Furthermore, at most one small item can be alone in a bin: $|A| \geq 2n - |L| - 1$. Thus, the performance ratio is

$$\frac{|A|}{|A|+|R|} \geq \frac{2n-|L|-1}{2n-|L|-1+n-|L|} \geq \frac{2n-1}{3n-1} = \frac{2}{3} - \frac{1}{9n-3}.$$

Case 2: $s \leq \frac{1}{2}$: Since we consider the competitive ratio on accommodating sequences, an optimal off-line algorithm, OPT, can pack all items in S. It may be instructive to view the optimal packing as being done in 3 phases:

1. UFF is run on S.
2. The packed items are rearranged, creating room for the rejected items.
3. The rejected items are packed.

The packing after Phase 1 is denoted by P_{UFF}, and the packing after Phase 3 is denoted by P_{OPT}. Similarly, E_{UFF} and E_{OPT} are used to denote the total empty space after Phase 1 and Phase 3, respectively. We assume without loss of generality that no large item is moved during Phase 2.

We use the following equation to bound the number of small items rejected.

$$|R_s| \leq \frac{1}{s} \cdot \left(E_{\mathrm{UFF}} - E_{\mathrm{OPT}} - \frac{|R_l|}{2} \right)$$

It is easy to see that $|R| < n$, since the empty space in any bin in P_{UFF} is less than s and all rejected items have size at least s. Thus, if all bins contain at least two items each, $\frac{|A|}{|A|+|R|} > \frac{2n}{2n+n} = \frac{2}{3}$, and we are through. Therefore, assume that some bins contain only one item. Since the empty space in any bin is less than $\frac{1}{2}$, such items must be large. Thus, the items that are alone in a bin are exactly the items in L.

It is now clear that $|A| \geq 2n - |L|$. However, if some bins contain more than two items, this lower bound is too pessimistic. Therefore, we try to "spread out" the items a little more. Assume that the items in P_{UFF} are labeled with consecutive numbers in each bin according to their arrival time, i.e., the first item in a bin is labeled 1, the next one is labeled 2, and so on. We split Phase 2 into two subphases, 2A and 2B, such that in Subphase 2A only items with labels higher than 2 are moved, and in Subphase 2B, the remaining moves are performed. Note that the pseudo-packing produced during Subphase 2A is only technical and used for counting purposes; it might not be a legal packing in that some bins might contain items with a total size larger than 1.

If some of the items which are moved during Subphase 2A are moved to bins containing items from L, a better lower bound on $|A|$ can now be obtained. The set of items that are still alone after Subphase 2A is divided into two sets: L_B, containing the items that are still alone after Subphase 2B, and L_A, containing those that are not. Any item that is alone after Subphase 2A was alone in P_{UFF} as well. Since no such item can be combined with an item belonging to R, each item in L_B is also alone in P_{OPT}. Therefore, the bins containing an item from L_B do not contribute to $E_{\mathrm{UFF}} - E_{\mathrm{OPT}}$.

Note that, since $L_A \cup L_B$ is the set of objects that are alone after Subphase 2A, $|A| \geq 2n - |L_A| - |L_B|$. The next lemma shows that increasing the number of items alone in P_{UFF} after Subphase 2A or 2B will increase the number of (large) items rejected due to admission control.

Lemma 4.1 $|R_l| \geq |L_A| + \frac{1}{2}|L_B| - 1$.

Proof Recall that t denotes the time just after the last large item was accepted by UFF and that A_t denotes the set of items accepted at time t. Since a large item was accepted just before time t, all items previously rejected are large items and therefore contained in R_l. Since the item was accepted, $\frac{|A_t|-1}{|A_t|-1+|R_l|+1} < \frac{2}{3}$. Solving for $|R_l|$, we get $|R_l| > \frac{1}{2}|A_t| - \frac{3}{2}$, and since $|R_l|$ must be an integer, we get $|R_l| \geq \frac{1}{2}|A_t| - 1$. We complete the proof by showing that $|A_t| \geq 2|L_A| + |L_B|$. To show this, we mark all small items accepted at time t, and to every item $o \in L_A$, we assign a unique marked item as described below. Since no item in L_A is alone after Phase 2, we can assume that the bin b_o containing o will receive at least one item, o', labeled 1 or 2 during Phase 2. If o' is marked, it is assigned to o. Otherwise, it must be labeled 2, since all items labeled 1 in bins before b_o are marked. The item which was packed below o' in P_{UFF} was alone at time t. Therefore, this item is not moved to any item in L_A. This item (labeled 1) can be assigned to o. In this way, every item in L_A has an item assigned which arrived before time t and which is not in $L_A \cup L_B$. Since $L_A \cup L_B \subseteq A_t$, $|A_t| \geq 2|L_A| + |L_B|$. $\square$

Subcase 2a: $s \leq \frac{1}{3}$. Since the smallest item in R has size s, the empty space in each bin in P_{UFF} is smaller than s. Thus, we can use $s(n - |L_B|)$ as an upper bound on $E_{\text{UFF}} - E_{\text{OPT}}$:

$$\begin{aligned} |R_s| &\leq \frac{1}{s} \cdot \left(E_{\text{UFF}} - E_{\text{OPT}} - \frac{|R_l|}{2} \right) < \frac{1}{s} \left(s(n - |L_B|) - \frac{|R_l|}{2} \right) \\ &= n - |L_B| - \frac{|R_l|}{2} \leq \; n - |L_B| \; - \frac{3}{2}|R_l|. \end{aligned}$$

Now, using Lemma 4.1, we get

$$|R| = |R_s| + |R_l| \leq n - |L_B| - \frac{1}{2}|R_l| \leq n - |L_B| - \frac{1}{2}\left(|L_A| + \frac{1}{2}|L_B| - 1\right)$$
$$= n - \frac{5}{4}|L_B| - \frac{1}{2}|L_A| + \frac{1}{2}.$$

Thus,

$$\frac{|A|}{|A|+|R|} \geq \frac{2n - |L_A| - |L_B|}{2n - |L_A| - |L_B| + (n - \frac{5}{4}|L_B| - \frac{1}{2}|L_A| + \frac{1}{2})}$$
$$\geq \frac{2n - (|L_A| + |L_B|) + \frac{1}{3}}{3n - \frac{3}{2}(|L_A| + |L_B|) + \frac{1}{2}} - \frac{\frac{1}{3}}{3n - \frac{3}{2}(|L_A| + |L_B|) + \frac{1}{2}}$$
$$\geq \frac{2}{3} - \frac{2}{12n-3},$$

since $|L_A| + |L_B| \leq \frac{2}{3}(n+1)$, which follows from the fact that the number of large items is at most n: $n \geq |R_l| + |L_A| + |L_B| \geq (|L_A| + \frac{1}{2}|L_B| - 1) + |L_A| + |L_B| \geq \frac{3}{2}(|L_A| + |L_B|) - 1$.

Subcase 2b: $\frac{1}{3} < s \leq \frac{1}{2}$. In this case, $s(n - |L_B|)$ is not a good bound on $E_{\mathrm{UFF}} - E_{\mathrm{OPT}}$, but we will show that even in this case, $E_{\mathrm{UFF}} - E_{\mathrm{OPT}}$ is "almost" bounded by $\frac{1}{3}(n - |L_B|)$, if $n \geq 9$ and $\frac{|A|}{|A|+|R|} < \frac{2}{3}$. Lemma 4.2 below with $c = 2$ is used for this purpose. It says that bins containing two or more items are filled to at least $\frac{2}{3}$ on the average.

Lemma 4.2 Let m be the number of bins containing at least c items in a First-Fit packing. If $c \geq 1$ and $m \geq c+1$, then the total size V of the items in these m bins is more than $\frac{c}{c+1}m$.

Proof Let C denote the set of bins containing at least c items, and, for any bin b, let $V(b)$ denote the total size of the items in b.

Suppose, for the sake of contradiction, that $V \leq \frac{c}{c+1}m$. Then there is a bin $b \in C$ such that $V(b) = \frac{c}{c+1} - \varepsilon$, $\varepsilon \geq 0$. The size of any item placed in a bin to the right of b must be greater than $\frac{1}{c+1}+\varepsilon$, since otherwise it would fit in b. Therefore any bin $b' \in C$ to the right of b has $V(b') > \frac{c}{c+1}+c\varepsilon \geq \frac{c}{c+1}$. This means that there is only one bin $b \in C$ with $V(b) \leq \frac{c}{c+1}$, and if b is not the rightmost nonempty bin in C, then $V > (m-2)\frac{c}{c+1}+(\frac{c}{c+1}-\varepsilon)+(\frac{c}{c+1}+c\varepsilon) \geq m\frac{c}{c+1}$. Thus, b must be the rightmost nonempty bin in C.

One of the items in b must have size at most $\frac{1}{c+1}-\frac{\varepsilon}{c}$. Since this item was not placed in one of the $m-1$ bins to the left of b, these must all be filled to more than $\frac{c}{c+1}+\frac{\varepsilon}{c}$. Thus, $V > (m-1)(\frac{c}{c+1}+\frac{\varepsilon}{c})+(\frac{c}{c+1}-\varepsilon) = m\frac{c}{c+1}+(m-1)\frac{\varepsilon}{c}-\varepsilon \geq m\frac{c}{c+1}+c\frac{\varepsilon}{c}-\varepsilon = m\frac{c}{c+1}$, which is a contradiction. $\square$

Assuming that $n \geq 9$, Lemma 4.2 combined with Lemma 4.3 below says that the average empty space in bins containing more than one item is at most $\frac{1}{3}$.

Lemma 4.3 Assume that $n \geq 9$ and $s \leq \frac{1}{2}$. Then, in P_{UFF}, at least three bins contain two or more items.

Proof Assume for the sake of contradiction that fewer than three bins contain at least two items. We count the number of large items. Since $s \leq \frac{1}{2}$, no bin contains a single item of size at most $\frac{1}{2}$. Therefore, at least $n-2$ bins contain large items, which all arrived before time t, i.e., $A_t \geq n-2$. By the first part of the proof of Lemma 4.1, at least $\frac{1}{2}A_t - 1$ large items are rejected. Noting that there can be at most n large items, we get $n-2+\frac{n-2}{2}-1 \leq n$. Solving for n yields $n \leq 8$, which is a contradiction. $\square$

Our goal is now, roughly speaking, to show that the average empty space in *all* n bins is bounded by approximately $\frac{1}{3}$. Let l be the number of the bin in which the last large item was placed. Let e denote the largest empty space in bins containing an item from L. In the proof of Lemma 4.5, we show a lower bound on the number of bins to the right of l of approximately $\frac{|L|}{2}$. Each of these bins contains at least two items of size larger than e. Thus, even if $e > \frac{1}{3}$, the average empty space in the L-bins and the bins to the right of l will be bounded above by approximately $(|L|\, e + (1-2e)(n-l)) \,/\, (|L|+n-l) \lesssim \left(|L|\, e + (1-2e)\frac{|L|}{2}\right) / \frac{3|L|}{2} = \frac{|L|}{2} \cdot \frac{2}{3|L|} = \frac{1}{3}$. Lemma 4.2 combined with Lemma 4.4 below says that we can assume that the rest of the bins have an average empty space of at most $\frac{1}{3}$.

Lemma 4.4 Assume that $n \geq 9$, $s \leq \frac{1}{2}$, $e \geq \frac{1}{3}$, and $\frac{|A|}{|A|+|R|} < \frac{2}{3}$. Then, in P_{UFF}, at least three of the first l bins contain two or more items.

Proof We count the total number of items of size larger than e. Since $|A| \geq 2n - |L|$, more than $n - \frac{|L|}{2}$ items are rejected, because otherwise we have a performance ratio of $\frac{2}{3}$, which would be a contradiction. To the right of bin l, there are $n-l$ bins containing at least two items each. All of the rejected items and those in the last $n-l$ bins are larger than e and there are more than $n - \frac{|L|}{2} + 2(n-l)$ of them. Bins containing items from L cannot accept any of these items, and only two can be put together since $e \geq \frac{1}{3}$. Thus, $n - \frac{|L|}{2} + 2(n-l) \leq 2(n-|L|)$. Solving for l, we get $l \geq \frac{n}{2} + \frac{3}{4}|L|$. This shows that at least $\frac{n}{2} - \frac{|L|}{4}$ bins to the left of l contain two or more items. By Lemma 4.3, $|L| \leq n-3$. Thus, $\frac{n}{2} - \frac{|L|}{4} \geq \frac{n}{2} - \frac{n-3}{4} = \frac{n+3}{4} \geq 3$, since $n \geq 9$. $\square$

Lemma 4.5 Assume that $n \geq 9$, $s \leq \frac{1}{2}$, and $\frac{|A|}{|A|+|R|} < \frac{2}{3}$. Then, $E_{\mathrm{UFF}} - E_{\mathrm{OPT}} < (n - |L_B|)\frac{1}{3} + \frac{1}{2}$.

Proof In the case where $e \leq \frac{1}{3}$, we have an upper bound of $\frac{1}{3}$ on the average empty space in every bin. Thus, $E_{\mathrm{UFF}} - E_{\mathrm{OPT}} \leq (n - |L_B|)\frac{1}{3}$. Now, assume that $e > \frac{1}{3}$. First we show an upper bound on l. At time t, no two bins can contain only one small item each. Therefore, $|A_t| \geq 2l - |L| - 1$. The total number of large items is at least $|R_l| + |L| \geq \frac{1}{2}|A_t| - 1 + |L| \geq l + \frac{|L|}{2} - \frac{3}{2}$. Since OPT must pack all these items in separate bins, we have $l + \frac{|L|}{2} - \frac{3}{2} \leq n$. Define $z \geq 0$ such that $n - l = z + \frac{|L|}{2} - \frac{3}{2}$. Since every bin to the right of bin l has two items of size greater than e, we have the following upper bound on the empty space in these $n - l$ bins and the bins with an item from $L \setminus L_B$: $e(|L| - |L_B|) + (1-2e)(n-l) = e|L| - e|L_B| + (1-2e)(z + \frac{|L|}{2} - \frac{3}{2}) < e|L| - \frac{|L_B|}{3} + (1-2e)\frac{|L|}{2} + (1-2e)(z - \frac{3}{2}) = \frac{|L|}{2} - \frac{|L_B|}{3} + (1-2e)(z - \frac{3}{2}) \leq \frac{|L|}{2} - \frac{|L_B|}{3} + (1-2e)z < \frac{|L|}{2} - \frac{|L_B|}{3} + \frac{1}{3}z$. Among the remaining bins, $l - |L| = n - z - \frac{3|L|}{2} + \frac{3}{2}$ bins do not contain an item from L_B. All of these bins have at least two items, and according to Lemma 4.4, enough of these bins exist for us to conclude, by Lemma 4.2 (with $c = 2$), that the empty space is at most $\frac{1}{3}(n - z - \frac{3|L|}{2} + \frac{3}{2})$. The total empty space is then less than $\frac{|L|}{2} - \frac{|L_B|}{3} + \frac{z}{3} + \frac{1}{3}(n - z - \frac{3|L|}{2} + \frac{3}{2}) = \frac{1}{3}(n - |L_B| + \frac{3}{2})$. □

Then, by Lemma 4.5, if $n \geq 9$,

$$
\begin{aligned}
|R_s| &\leq \frac{1}{s} \cdot \left(E_{\mathrm{UFF}} - E_{\mathrm{OPT}} - \frac{1}{2}|R_l| \right) < \frac{1}{s} \left(\frac{1}{3}(n - |L_B|) + \frac{1}{2} - \frac{1}{2}|R_l| \right) \\
&\leq n - |L_B| + \frac{3}{2} - \frac{3}{2}|R_l|.
\end{aligned}
$$

Using Lemma 4.1 as in Subcase 2a, we get that, for $n \geq 9$,

$$
|R| < n - |L_B| + \frac{3}{2} - \frac{1}{2}(|L_A| + \frac{1}{2}|L_B| - 1) = n - \frac{5}{4}|L_B| - \frac{1}{2}|L_A| + 2.
$$

Thus,

$$\begin{aligned}
\frac{|A|}{|A|+|R|} &\geq \frac{2n-|L_A|-|L_B|}{2n-|L_A|-|L_B|+(n-\frac{5}{4}|L_B|-\frac{1}{2}|L_A|+2)} \\
&\geq \frac{2n-|L_A|-|L_B|}{3n-\frac{3}{2}|L_A|-\frac{9}{4}|L_B|+2} \\
&\geq \frac{2n-(|L_A|+|L_B|)+\frac{4}{3}}{3n-\frac{3}{2}(|L_A|+|L_B|)+2} - \frac{\frac{4}{3}}{3n-\frac{3}{2}(|L_A|+|L_B|)+2} \\
&= \frac{2}{3} - \frac{4}{6n+3}, \text{ for } n \geq 9.
\end{aligned}$$

This bound is lower than the lower bounds obtained in Case 1 and Subcase 2a for all n. □

For completeness we show that this bound is asymptotically tight:

Theorem 4.2 Unfair-First-Fit has a competitive ratio of at most $\frac{2n}{3n-1}$ on accommodating sequences.

Proof Let $\varepsilon > 0$ be small enough. The adversary can give the following sequence:

- n pairs: $\frac{1}{3}+(n-i)\varepsilon$ and $\frac{1}{3}+i\varepsilon$, for $i=1,\ldots,n$.
- $n-1$ items of size $\frac{1}{3}-(n-1)\varepsilon$.

Unfair-First-Fit does not reject any items due to admission control, since the items are all smaller than $\frac{1}{2}$ by the choice of ε. Therefore it will pack only the first $2n$ elements, whereas OPT can behave as First-Fit, except on the very first element which is placed in the last bin. The performance ratio is $\frac{2n}{3n-1}$. □

Besides showing that the proof above is asymptotically tight, the sequence also shows that if a ratio better than $\frac{2}{3}$ should be obtained, admission control should be performed on items of size significantly smaller than $\frac{1}{2}$.

5 Comments on Other Possible Algorithms

5.1 Unfair-Any-Fit

As a final comment on Unfair-First-Fit, we discuss the following question:

> Is the choice of First-Fit as the packing algorithm important or could an arbitrary Any-Fit unfair algorithm be used in its place, still with the same admission control?

We show a negative answer to this question by giving a sequence on which an Unfair-Any-Fit variant would have a competitive ratio on accommodating sequences of $\frac{12}{19-\frac{7}{n}}$ which is below say 0.64 for $n > 28$.

Lemma 5.1 There exists an Any-Fit algorithm $\mathbb{A}$ such that when $\mathbb{A}$ is used instead of First-Fit in the algorithm Unfair-First-Fit, the competitive ratio is at most $\frac{12}{19-\frac{7}{n}}$, for n divisible by 7.

Proof The Any-Fit algorithm we consider does the following: Whenever the item can be packed within the bins already open, it uses the bin which was opened last. If there is no space in the open bins, the algorithm opens a new bin (if there is one), otherwise the item is rejected. Let $n = 7l$ for $l \in \mathbb{N}$, and let $\varepsilon > 0$ be small enough. An adversary could give the following sequence:

- $2l$ times the triple $\langle \frac{1}{2} - 2\varepsilon, 4\varepsilon, \frac{1}{2} + \varepsilon \rangle$.
- l times the triple $\langle \frac{1}{2} + 2\varepsilon, \frac{1}{2} + 2\varepsilon, \frac{1}{2} + \varepsilon \rangle$.
- $3l$ times the pair $\langle \frac{1}{2} - \varepsilon, 3\varepsilon \rangle$.

- $4l - 1$ times $\frac{1}{2}$.

From the first $2l$ triples, $\mathbb{A}$ packs the two first items in one bin. The third item in the triple is rejected due to admission control. This uses $2l$ bins. Next, all items of size $\frac{1}{2} + 2\varepsilon$ are accepted, whereas items of size $\frac{1}{2} + \varepsilon$ are rejected due to admission control. This will use another $2l$ bins. Next, the $3l$ pairs will be packed in the remaining $3l$ bins, one pair in each bin. All items of size $\frac{1}{2}$ must be rejected, since the empty space in each bin is less than $\frac{1}{2}$. The total number of accepted items is $2 \cdot 2l + 2l + 2 \cdot 3l = 12l$. The total number of items given is $3 \cdot 2l + 3l + 2 \cdot 3l + 4l - 1 = 19l - 1$, resulting in a performance ratio of $\frac{12l}{19l-1} = \frac{12}{19-\frac{7}{n}}$.

We now show how OPT can pack the entire sequence: Items of size $\frac{1}{2} - 2\varepsilon$ and $\frac{1}{2} + 2\varepsilon$ are packed together and items of size $\frac{1}{2} - \varepsilon$ and $\frac{1}{2} + \varepsilon$ are packed together. This will use a total of $5l$ bins. The items of size $\frac{1}{2}$ can be packed in $2l - 1$ bins plus half of the last bin. Finally, small items (of size 3ε and 4ε) have a total size of $17l\varepsilon < \frac{1}{2}$ for small enough ε. Hence these items are packed in the remaining part of the last bin. □

5.2 Randomized Algorithms

Finally, we would like to comment on a simple way to convert an algorithm for the Classical (minimization) Bin Packing problem to a randomized algorithm for the Unrestricted (maximization) Bin Packing problem on accommodating sequences. Assume that we are given an algorithm $\mathbb{A}$ that is known to be able to pack any accommodating sequence of items in βn bins for some constant β. We can simulate this algorithm using βn "virtual" bins. At the beginning, the randomized algorithm $\mathbb{R}$ randomly decides which n of the βn virtual bins are going to correspond to the "real" n bins. If the simulation of $\mathbb{A}$

packs an item in a bin that corresponds to a real bin, then $\mathbb{R}$ packs it in the corresponding real bin. All other items are rejected. The expected fraction of the items which $\mathbb{R}$ accepts is at least $1/\beta$, since on average $\frac{n}{\beta n} = \frac{1}{\beta}$ of the items accepted by $\mathbb{A}$ are packed. The algorithm with the best known value of β is Harmonic++ [13]. In [13], it is shown that when n goes to infinity, β goes to a value that is at most 1.58889. This yields an algorithm for Unrestricted bin packing with a competitive ratio of about $\frac{1}{1.58889} \approx 0.629$. This is slightly better than the performance of First-Fit (0.625), but worse than that of Unfair-First-Fit (0.666). It is also shown that HARMONIC++ belongs to a class of algorithms called SUPER HARMONIC and that no algorithm in this class has a β smaller than 1.58333. Furthermore, it is proven in [15] that no online algorithm can have a β smaller than 1.54014. Thus, using this approach we cannot get a competitive ratio better than $\frac{1}{1.54014} \approx 0.649$, which is worse than the performance achieved by our Unfair-First-Fit.

Acknowledgment

We would like to thank two anonymous referees for many useful comments which have definitely improved the presentation of our results. Regarding extra results, the sequence in Theorem 4.2 was found by one of the referees who also raised the question regarding Any-Fit which was answered in Lemma 5.1.

References

[1] B. Awerbuch, Y. Azar, and S. Plotkin. Throughput-Competitive On-Line Routing. In *34th IEEE Symposium on Foundations of Computer Science*, pages 32–40, 1993.

[2] B. Awerbuch, Y. Bartal, A. Fiat, and A. Rosén. Competitive Non-Preemptive Call Control. In *Proc. 5th Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 312–320, 1994.

[3] B. Awerbuch, R. Gawlick, T. Leighton, and Y. Rabani. On-line Admission Control and Circuit Routing for High Performance Computation and Communication. In *35th IEEE Symposium on Foundations of Computer Science*, pages 412–423, 1994.

[4] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis.* Cambridge University Press, 1998.

[5] J. Boyar and K. S. Larsen. The Seat Reservation Problem. *Algorithmica*, 25:403–417, 1999.

[6] J. Boyar, K. S. Larsen, and M. N. Nielsen. The Accommodating Function: A Generalization of the Competitive Ratio. *SIAM Journal on Computing.* To appear. Preliminary version in *Sixth International Workshop on Algorithms and Data Structures*, volume 1663 of *Lecture Notes in Computer Science*, pages 74–79. Springer-Verlag, 1999.

[7] E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson. Approximation Algorithms for Bin Packing: A Survey. In Dorit S. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*, chapter 2, pages 46–93. PWS Publishing Company, 1997.

[8] E. G. Coffman, Jr., J. Y-T. Leung, and D. W. Ting. Bin Packing: Maximizing the Number of Pieces Packed. *Acta Informatica*, 9:263–271, 1978.

[9] E. G. Coffman, Jr. and Joseph Y-T. Leung. Combinatorial Analysis of an Efficient Algorithm for Processor and Storage Allocation. *SIAM J. Comput.*, 8:202–217, 1979.

[10] J. Csirik and G. Woeginger. On-Line Packing and Covering Problems. In Amos Fiat and Gerhard J. Woeginger, editors, *Online Algorithms*, volume 1442 of *Lecture Notes in Computer Science*, chapter 7, pages 147–177. Springer-Verlag, 1998.

[11] D. S. Hochbaum, editor. *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company, 1997.

[12] Anna R. Karlin, Mark S. Manasse, Larry Rudolph, and Daniel D. Sleator. Competitive Snoopy Caching. *Algorithmica*, 3:79–119, 1988.

[13] S. Seiden. On the Online Bin Packing Problem. To appear in Proceedings of the 28th International Colloquium on Automata, Languages and Programming, July 2001.

[14] Daniel D. Sleator and Robert E. Tarjan. Amortized Efficiency of List Update and Paging Rules. *Comm. of the ACM*, 28(2):202–208, 1985.

[15] A. van Vliet. An Improved Lower Bound for Online Bin Packing Algorithms. *Information Processing Letters*, 43:277–284, 1992.